AD-A055 080 NAVAL SURFACE WEAPONS CENTER DAHLGREN LAB VA F/G 9/2

NAVY DISTRIBUTED COMPUTER SYSTEMS WORKSHOP (JUNE 1977).(U)

APR 78 W P WARNER, W J DEJKA

UNCLASSIFIED NSWC/DL-TR-3790 NL

1 OF 1

AD
A055080

END
DATE
FILMED
7-78
DDC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NSWC/DL-TR-3790 ~~TR-3790~~	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NAVY DISTRIBUTED COMPUTER SYSTEMS WORKSHOP (JUNE 1977).		5. TYPE OF REPORT & PERIOD COVERED Final rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Walter W. P./Warner ~~NSWC, Dahlgren, VA~~ William W. J./Dejka ~~NOSC, San Diego, CA~~		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center Dahlgren Laboratory (K70) Dahlgren, VA 22448		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NIF
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Surface Weapons Center Dahlgren Laboratory Dahlgren, VA 22448		12. REPORT DATE April 1978
		13. NUMBER OF PAGES 48 p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

distributed computer systems
microcomputer
systems designs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Results of a workshop held for the purpose of discussing the impact, payoff, and technical issues of distributed computer systems are reported. At the workshop, attendees with diverse backgrounds and experience were able to (1) have a common understanding of distributed systems designs criteria; (2) conduct a quick and complete review of distributed concepts and approaches used in current Navy systems implementation, (3) conduct a state-of-the-art survey of distributed systems theory, (4) compare common issues in relation to

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391598

78 06 14 012

20. ABSTRACT (Continued)

Navy-distributed systems, and (5) provide discussions beneficial to both technical and management areas on trends, ideas, and technologies in Navy systems development.

FOREWORD

The workshop, "Distributed Computer Systems," was held in Fredericksburg, Virginia, 7-9 June 1977 and sponsored by the Chief of Naval Material* for the purpose of discussing the impact, payoff, and technical issues of distributed computer systems. The concept of distributing the computing portion of systems is the result of new advances in component and device technology, particularly the microcomputer; and the Navy has a number of programs implementing these concepts as an integral part of systems designs.

This report consists of reviews by the authors and, as such, constitutes their understanding of the talks presented. The authors would like to apologize for any errors or misinterpretations and request that these not be considered as a reflection on the presentors.

This report was reviewed and approved by Walter P. Warner, Head, Computer Program Division, Strategic Systems Department, Naval Surface Weapons Center, Dahlgren, Virginia; and William J. Dejka, Naval Ocean Systems Center, San Diego, California.

Released by:

R. A. NIEMANN, Head
Strategic Systems Department

* NAVMAT ltr 1 Apr 1977 031:AJD

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

INTRODUCTION

OBJECTIVES

The principal topics of the workshop were the design concepts and implementation approaches for exploiting low-cost hardware and simplifying software through distributed systems. Particular goals included were to:

a. Provide a common understanding of distributed systems design criteria and program objectives influenced by distributed computer systems concepts

b. Conduct a quick but complete review of distributed concepts and approaches used in current Navy systems implementations

c. Present a survey of the state of the art of distributed systems theory and ongoing research

d. Identify common issues as they apply/relate to Navy distributed systems and define additional areas of research and development

e. Provide a forum for the discussion of trends, ideas, and technologies, as well as impact or potential payoff which can be mutually beneficial to both technical and management areas

This workshop was intended to meet the needs of many with diverse backgrounds and experience, and it reflects the importance of this complex subject in future Navy systems development.

ISSUES

Many different issues were identified as critical or as having an impact on distributed systems designs:

1. Definitions of distributed computing, distributed systems, and other basic terms have many different meanings to various individuals; and a complete set of accepted definitions is critical to fostering technology, through communications, both with the written word or personal interaction.

2. Standards in bussing, protocol, software, and other design and specification areas are critical to the quick and effective design of systems. These standards must not limit future enhancement of technology however.

3. Whether to have autonomous nodes or centralized control is an issue which must be resolved based on the practicality of design in both the near and far term.

4. The choice of dedicated function processors or total reconfigurability is an issue which separates into the technology for large general-purpose computer designs and the distribution of large numbers of functionally dedicated microcomputers.

5. Techniques must be developed for systems trade-offs and performance estimation in view of rapidly changing design requirements.

6. Operational implications, requirements/risk impact, etc., will be concerns if the Navy widely accepts new distributed systems design techniques.

7. Interunit communication involving levels of protocol and unit synchronization are necessary, but their identifications and definitions must be undertaken.

8. Security of systems under distributed and dedicated computer systems design and its criticality in meeting future system requirements are necessary.

9. Fault-tolerance and testability issues, maybe the single most difficult and complex areas associated with distributed systems designs, must be addressed through a planned program of research and development.

10. Software continues to be a cost-driving factor in systems designs and will continue with distributed systems.

FORMAT

In order to meet the objectives of the workshop, the format chosen for the meeting was to include:

1. Introductory session to provide a state-of-the-art overview

2. Three technical sessions consisting of six 10-min presentations followed by a 2-hr open discussion

3. Summary session with eight 10-min summaries followed by 2 hr of open discussion. The agenda for the workshop is given in Table 1.

Table 1. Workshop Agenda

7-9 June 1977
Holiday Inn North
Fredericksburg, Va.

Tuesday, 7 June 1977

Session 1

0900	Welcome - Commander, NSWC, Dahlgren
0910	Overview - Jack Dietzler, MAT 031
0930	Keynote address - CAPT Richard V. Wilson, NAVAIR 533
0945	Taxonomy of Distributed Systems - Doug Jensen, Honeywell
1015	Break
1030	Definition of Distributed Systems - Dr. Phillip Enslow, Georgia Institute of Technology
1100	Operating Systems Issues - Charles Arnold, NUSC
1130	Survey of Current Multicomputer Research - Dan Siewiorek, Carnegie-Mellon University
1200	Lunch

Session II

1300	Distributed Computer Systems Goals for Shipboard Command Control Applications - D. G. Mudd, NOSC, Code 733
1310	Surface Combatant Ship Combat Systems Architecture - R. P. Cullen, NSWC
1320	The Application of Structures Design and Distibuted Techniques to Avionics Information Processing Architectures - Louis A. Naglak, Information Processing and Display Division, NADC
1330	SEAMOD - Don Eddington, NOSC
1340	Interconnection Technology and Navy Systems - Allan Clearwaters, NUSC, New London
1350	Overview of Research Programs in Distributed Computing - Leonard S. Haynes, ONR
1400-1600	Open Discussion
1700	Cocktail Hour
2000	Panel Discussion - Second Thoughts on Distributed Systems - J. Shore, A. VanDamm, Y. S. Wu, W. Dejka

Wednesday, 8 June 1977

Session III

0900	A Command Control Concept - John Griffin, EG&G
0910	Distributed Computer Investigations for Shipboard Command Control Systems Applications - E. J. Wells, NOSC
0920	Avionic Information Processing Systems Designs Methodology - Stanley B. Greenberg, NADC
0930	Monitoring a Distributed Processing System Richard Fryer, NWC
0940	A Multi-Distributed Processor System (MDPS), William Sheppard, NOSC
0950	Flow graphs and Data Flow graphs Used to Aid the Partitioning Process, Uno R. Kodres, NPGS
1000-1200	Open Discussion
1200	Lunch

Session IV

1300	SECNAV Policy guidance, B. Zempolich, NAVAIR
1310	Protocols, Paul Levine, NUSC, New London
1320	Adaptable Shipboard Tactical Data Distribution System, LCDR H. C. Schleicher, NOSC
1330	MIL-STD 1553
1340	IEEE Standard 488 - Professor Jim Howard, UCSB
1350	National I/O Interface Standards, Dell Shoemaker, GSA
1400-1600	Open Discussion: MIL-STD 1553 - Richard DeSipio, NADC Buses and Distributive Computer Systems - David Rennels, JPL A Family of Special-Purpose Processors For Distributed Dedicated Computer Systems - Maniel Vineberg, NOSC

Thursday, 9 June 1977

Session V

0900-1000	Group Discussion
1000-1100	Open Discussion
1100-1200	Overall Summary

This format resulted in maximum interaction between participants. Each technical presentation, limited to 10 min, resulted in concise, to-the-point viewgraphs focusing on task objective, approach, and summary of results. Any further information on the subject matter was presented during the 2-hr open-discussion period and only after the participants strongly requested amplifying data.

PARTICIPANTS

Various Navy organizations and many different levels of technical and managerial personnel were participants at the workshop (see Appendix A). A summary of the representation by organization is shown in Table 2.

Table 2. Organizational List of Attendees

ASN (R&D)	1	NOSC	12
OPNAV	1	NWC	2
NAVSEA	4	NAFI	3
NAVSUP	3	NTEC	1
NAVSEC	2	NCSL	1
NAVELEX	6	NRL	2
NAVMAT	4	NPGS	2
NAVAIR	2	AFSC	1
NAVFACENGCOMHQ	2	RADC	1
ONR	1	FACSO	1
NAVDAC	3	JPL	1
NUSC	10	CMC	1
NADC	14	CONTRACTORS	16
NSWC	31	TOTAL	128

OVERVIEW BY THE CHAIRMEN

The Distributed Computer Systems Workshop, sponsored by LCDR Jack Dietzler (MAT 0312), brought together people in the Navy who are working in the field. The purpose was to see if an agreement could be reached as to where the Navy stands today, what direction it is going in, and what issues the Navy R&D community faces in the field of distributed systems. CAPT Richard Wilson (NAVAIR 533) set the tone of the workshop by discussing some of the past, current, and future problems in the Navy's use of computers.

In some ways the workshop raised more questions than it settled as there were disagreements among the participants in many areas. It definitely served the purpose of bringing some of these issues to the forefront.

The introductory talks by Doug Jensen and Phil Enslow addressed the questions of definitions and categorizing distributed systems. There was disagreement over the definitions, and this was pointed out as one of the major hurdles to get over if the Navy was going to be able to come up with a reasonable set of standards. The question of standards was introduced by Mr. James Campbell of the Office of the Secretary of Navy.

Dan Siewiorek discussed present research in multicomputer systems, and other participants presented systems under development in the Navy.

Most of the problems seemed to center around how to organize a distributed system. One group was talking about distributing functions to a separate microcomputer, while another group was talking about organizing the microcomputers into a virtual uniprocessor system. Another issue raised was how to control a distributed system--whether one processor should be in total control at any given time, or whether control should be partitioned among the several processors.

One of the gaps evidenced was the availability of tools and techniques for conducting systems trade-offs and performance estimation.

Some interesting technical areas included the XDP, a distributed processor; the SEAMOD concept for distributing shipboard electronics; the use of flow graphs for systems partitioning; the selection of building blocks which can be distributed for fault-tolerance computing; and the development of a software language for computer systems monitoring. Without a doubt, the Navy demonstrated a clear understanding of the technology and, more importantly, a desire to solve the critical problems of the future.

The special session of bus protocol and interface standards was led by Mr. Bernie Zempolich and Dr. Bob Gordon and focused on the need for bus protocol standards for intersystem compatibility. The discussions were very interesting, and further efforts will be needed before the issues can be sifted out. A particularly interesting presentation was made by Dr. Jim Howard on the IEEE 488 B bus standard and its possible use in a microcomputer net.

The summary of the workshop included many technical areas:

1. There is a need for the Navy to clarify the definitions.

2. The software issues of a large multicomputer system need to be addressed in detail.

3. There is existing work, both in the universities and industry, focused on multicomputers but little or no work on distributed dedicated functions (SEAMODS is an exception).

4. The fault-tolerance and testing issues are extremely important, but the workshop did not address them on any level of significance.

The workshop was a success. It brought together 128 persons involved in some way with distributed systems. The participants spoke freely about the problems as they saw them. The state of the art, systems objectives, design approaches, and issues in distributed systems were discussed. Not too many issues were resolved, however.

Recommended areas for further resolution are:

1. Generally accepted definitions and taxonomy including executive/ operating systems functions followed by a good set of standards

2. The implications of future tactical operating philosophy to distributed systems designs considerations

3. Designs for reliability and maintainability--a critically important area achievable through distributed designs

It has often been stated that distributed systems designs reduce the life cycle of software, but the workshop demonstrated a cautious optimism in accepting this premise.

KEYNOTE ADDRESS
CAPT Richard V. Wilson
(NAVAIR 533)

The single most prominent problem faced by the program manager who is planning to base his system on advanced technology is the selection of the technology (or technologies). R&D development cycles, Figure 1, are becoming so compressed that the program manager must resort to using a crystal ball to predict the technologies which will be commonplace in the production time frame of his system. This problem is further aggravated by the tendency for old technologies to go out of production once they have been surpassed. In the electronics field we have this problem in "spades."

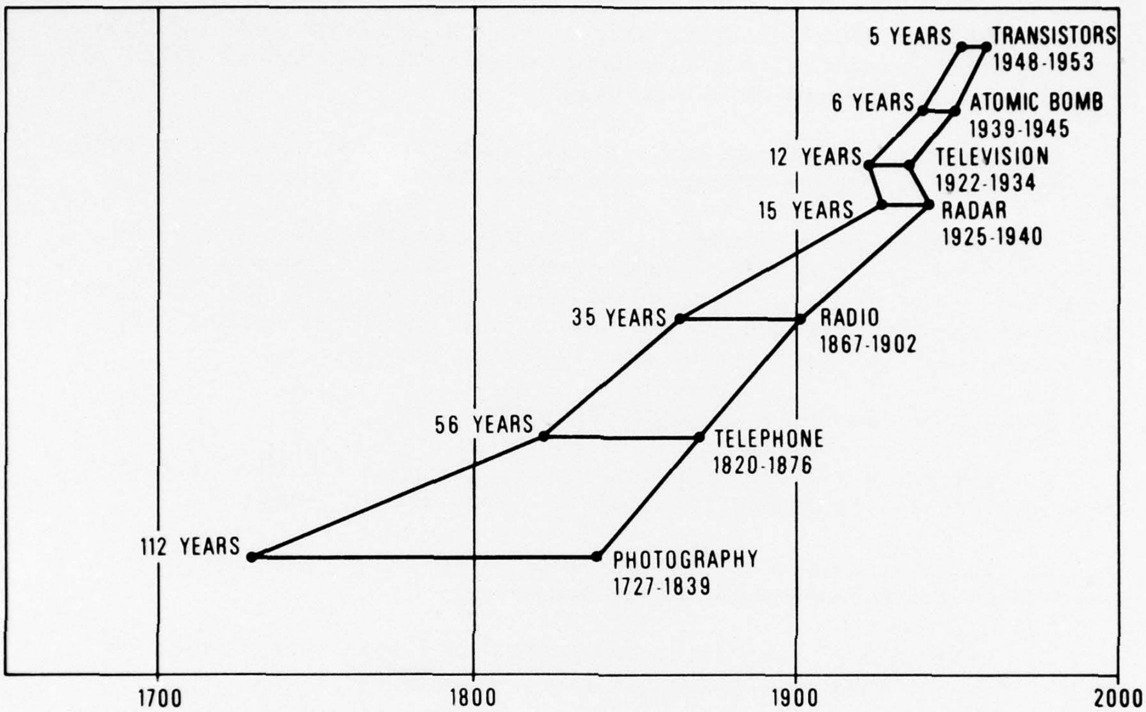


Figure 1. Research and Development Life Cycle

If the ordinate of Figure 1 is extended, technology would move in rapid succession from the transistor to the integrated circuit to the large-scale integrator circuit or LSI. Using LSI technology, the central computing unit of a minicomputer can be reduced to a single square piece of silicon (or sapphire), 100 or 200 mils on a side. Since the early 1970's, American industry has refined this process to the point where today it has a 16-bit minicomputer with a cycle time of 125 nsec in a single dual-in-line package. That LSI chip may dissipate a mere 100 mW of power; thanks to new sapphire-substrate technology. If these figures seem extreme to anyone in the audience, he should read the feature story in the latest issue of "Electronics" magazine.

The P-3C avionics system was designed in the 1960's to take advantage of an expensive computer resource. All primary tasks are channeled through the UNIVAC 1830, one of the most advanced airborne computers of that time. This was the Navy's first attempt to produce such a complex, computer-controlled ASW aircraft weapon system.

The experience of the 1960's resulted in a natural reaction by the engineering community to provide additional distributed-processing resources. These resources, for whatever the reason, were called by every name under the sun but "computers."

The S-3A (the next-generation ASW aircraft), for example, uses more than half a dozen such distribution "processors" in concert with its general purpose central computer--the AYK-10, an airborne version of the UYK-7.

By the early 1970's, the full realization of the cost to develop and utilize the myriad of computers used in tactical applications began to sink in. The result was a series of computer standardization programs and the creation of a Tactical Digital System Office at the Chief of Naval Material (CNM) staff level. This time frame also brought us to the production of the F-14 aircraft in the Navy. The design of the F-14, which embodies four programmable computers of varying complexity, utilized the concept of centralized analog-to-digital conversion. (The P-3 and S-3, by comparison, used distributed analog-to-digital conversion where needed.)

Thus was born the Computer Systems Data Converter (CSDC). There are a great number of analog inputs to this "box."

The CSDC is complex; there are over 5000 microcircuits in this "Converter." The modules are labeled "Computer" and "Memory." There is no data bus in the F-14 since the analog outputs of the various systems must first be "converted" by the CSDC before they can be used by the central computer. It is interesting to note that the F-14 cannot fly if the CSDC is not functioning.

By the mid 1970's, technology was well on its way toward imposing the use of a set of standard computer hardware and software on the Navy user. On the shipboard side of the house, the Navy had (and still has) the UYK-7 and UYK-20. The air side is the AYK-14. Still, the use of nonstandard hardware and software could not be restricted for sound technical and economic reasons.

The F-18, which does have a data bus and depends more on the use of digital subsystems, will use a pair of AYK-14 standard minicomputers. Two central computers, although not identical in capabilities (one has twice the memory capacity of the other), give the aircraft a degree of redundancy to ensure flight safety. In addition to the central mission computers, the aircraft will employ other embedded computing resources. Curiously, these computing resources will be assembled using mostly the AMD 2900 LSI microprocessor at their core.

The AYK-14, the first preproduction model of which was delivered to the Navy in May 1977, also uses the AMD 2900. So, in a sense, there is a measure of commonality at the LSI device level. More importantly, the use of the same LSI device by many vendors demonstrates the dependence of the aerospace industry on commercially developed LSI to economically build its electronic subsystems. For this reason, it is incumbent upon the users to find ways to facilitate the use of this commercial-technology base.

This must be done without reviving the problems of the 1960's and early 1970's.

Reviewing the situation, the following is revealed:

1. Phase I (1960's). Expensive data processing resources were used to their limit. This resulted in complex and possibly even more expensive software.

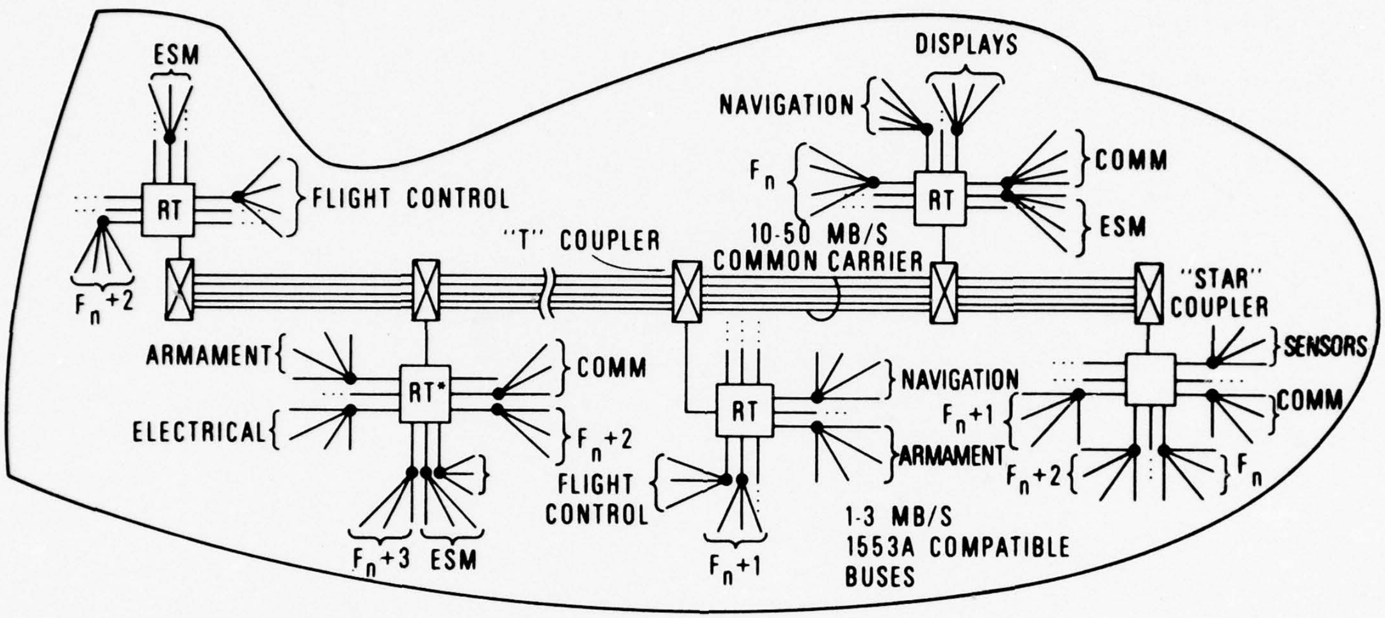
2. Phase II (early 1970's). The engineering community discovered distributed processing and used it to maximize performance as well as to create a great number of diverse processors.

3. Phase III (mid 1970's). Standards designed to solve the problems developed in Phase I and intended to curb the proliferation of Phase II were imposed.

4. Phase IV (the present). LSI-based microprocessors which are being rapidly introduced appear to solve the problems of Phases I and II and which, in turn, may force reassessment of the premise of our computer standardization program.

If the cost of computer hardware can be brought low enough where computers can economically replace simple functions (which, in fact, has already happened), then functions can be partitioned to the point where even software costs can be made to approach problem-definition costs. When this happens, the greatest challenge will lie in defining and developing the communications among and synchronization of advanced distributed networks of function-oriented elements.

In Figure 2, a potential system architecture for a 1985-1900 time-frame aircraft is shown. The communications among the regional terminals appear as its most distinguishing feature. Note there is no main central processor.



*RT = REGIONAL TERMINAL

Figure 2. Distributed Information Handling System Architecture

This last architecture may seem, at first glance, far afield of traditional partitions for similar systems, but revolutionary concepts must be considered within the context of a technology revolution. It is incumbent on each program manager to think in new and innovative ways of using the rapidly advancing technology. He must not be left behind; the commercial world is moving ahead.

TAXONOMY OF DISTRIBUTED SYSTEMS

E. Douglas Jensen (Honeywell)

Figure 3 presents a taxonomy, or naming scheme, for systems of interconnected computers. It is an attempt to provide an implementation-independent method by which to identify designs, and a common context in which to discuss them. The taxonomy is based on interprocessor message handling and hardware interconnection topology, and distinguishes ten basic multiple-computer architectures. Various relevant attributes are identified and discussed, and examples of actual designs are given for each architecture.*

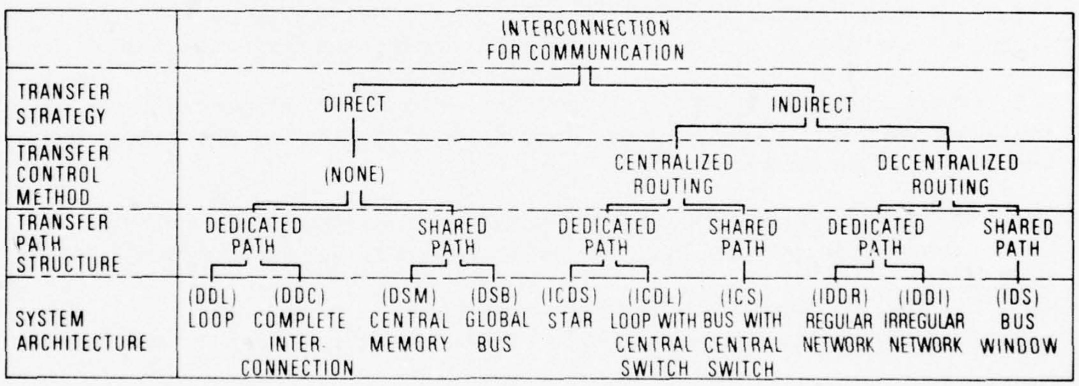


Figure 3. The Taxonomy

All taxonomy schemes, whether used by biologists, social scientists, or computer scientists, provide (1) a useful grouping according to distinctions and (2) a consistent, universally understood nomenclature. A taxonomy of computer systems has a third purpose; it can synthesize characteristics into something new to meet certain requirements. Such a system may be able to help in the development of a systematic design methodology, or even a theory, for distributive processing systems.

---

* George A. Anderson and E. Douglas Jensen, *Computer Interconnection Structures: Taxonomy, Characteristics, and Examples*, Computing Surveys, Vol. 7 No. 4 December 1975.

This type of taxonomy is called synthetic, as opposed to the purely descriptive nature of the other types. A synthetic taxonomy is defined as a space with points around it, like an N cube. It has none of the hierarchical problems of outlines and trees.

DEFINITION OF DISTRIBUTED SYSTEMS

Dr. Phillip Enslow (Georgia Institute of Technology)

The term "distributed processing" is being used with so many different connotations that the term itself has become meaningless. Sales people, especially, abuse the term by using it as a selling point. A good part of any conversation about distributed systems is spent in defining the term. These communication problems are serious enough to make the whole definition issue a management issue.

At this stage, the important point is not the definition itself, but rather the acceptance of the fact that a definition is required.

Distributed data processing is more than a new technique; it is an entirely new design philosophy. Specific definitional issues are:

1. Component Multiplicity. Functions can be reassigned and redistributed throughout the system. Any type of dedicated functionality is not distributed processing.

2. Component Distribution. Components are interconnected by a network. There are no master-slave relationships, either physically or logically.

3. System Unity. Some type of overall operating system must exist.

4. System Use. Services are requested by name, rather than by processing module which would effect a master-slave relationship.

5. Component Autonomy. A request can be refused, leading to bidding, and, perhaps, preemption.

Excluded from this definition are:

1. Simple and intelligent terminal systems
2. Front-end processors
3. Fragmented or dedicated functions
4. Master-slave relationships

The Bank of America operates a distributed processing system that meets the definition presented here; it also has a distributed data base, which is not essential to the definition. If one of the system's minicomputers was disconnected from the system, all other processes in the system would remain up without any software changes, except functions related to the data base contained in that processor. The processor itself would not be missed by the others.

OPERATING SYSTEM ISSUES FOR DISTRIBUTED SYSTEMS

Charles Arnold (NUSC)

Although distributive systems definitely have an important place in our technology, there is a danger of their being oversold. Along with this may come a second wave of software problems similar to what the Navy has just been through for centralized systems. We must step back from the development stage and examine some issues before they get out of hand.

Four key issues deserving particular attention are:

1. Fault tolerance

2. Protection and security. To protect jobs from each other and to prevent a hardware or software failure from bringing down an entire system, distributive systems need global as well as local operators.

3. Communications and synchronization. The deadlock condition is not a local issue; it must be handled on a global basis.

4. Structure and kernel concepts

SURVEY OF CURRENT MULTICOMPUTER RESEARCH

Dan Siewiorek (Carnegie-Mellon University)

A survey of implemented multicomputer systems reveals the range of major architectural techniques currently in use. The systems presented here have been organized into two major groups, according to whether they are based on messages or shared memory.

MESSAGE-BASED SYSTEMS

1. ARPANET. A store and forward network with an interface message processor (IMP) that routes and forwards submessages through the network to another host

2. ALOHANET. A central broadcast network in which all satellites broadcast on a contention channel with error control. Host sorts the messages and transmits back to all satellites on a separate channel

3. ETHERNET. An extension of the ALOHA architecture, allowing multiple broadcasting and receiving by using taps between all stations

4. TANDEM 16 NONSTOP. Designed for distributed-data-base management. *Isolates failures to one processor* by using dual-ported I/O devices. An interprocessor communication bus transfers messages, or even data files, between processors.

SHARED-MEMORY SYSTEMS

1. Carnegie Multi-Miniprocessor (C.MMP). A 16-by-16 crosspoint switch, allowing 16 processors and 16 memory banks to dynamically interact

2. UC at Berkeley PRIME. A multiport-memory system, with thirteen 4-port memories, five processors, and each processor tied to eight memories. Reliability was the basic goal.

3. PLESSY 250. Multiple buses and CPUs totally and/or partially tied to multiport memories

4. SIFT. Designed for commercial aviation, uses off-the-shelf processor/memory pairs. Memories are multiported. The processors can rewrite local memory, but have read-only access to all other memories.

5. BBN PLURIBUS. Consists of a shared memory, processor buses (two processors per bus), bus couplers connecting the two, plus I/O buses and bus couplers to them. The system operates without interrupts.

6. Computer Module Clusters. The fundamental building block is a processor-memory pair, called a computer module. Processors share a memory address space by sending global addresses to a time multiplexed, intelligent bus controller called a K-map. A cluster consists of one K-map and up to 14 computer modules. Logical mapping over intercluster buses allows memory sharing across a system.

Research problems that must be addressed for any architecture design are:

1. System organization

2. Interprocessor control

3. Deadlocks

4. Small address space

5. Operating system primitives

6. Reliability

7. Problem decomposition

DISTRIBUTED COMPUTER SYSTEMS GOALS FOR SHIPBOARD COMMAND CONTROL APPLICATIONS

Don Mudd (NOSC)

The basic problem of the shipboard command and control (C^2) computing function is the integration of a large number of sensors, communications links, and weapons. Presently, the typical processing and architecture of a C^2 system is dedicated towards handling each type of interface separately--a modular software design. A comparison of characteristics in five currently used systems (NTDS, JPTDS, CGN-38, SSNX, and AEGIS) reveals:

1. A typical memory capacity is approximately 500K 16-bit words.

2. The two largest systems require almost six times the processing speed of a UYK-20.

3. Output ranges from 42 to 128 16-bit channels.

4. Individual processing functions occurring in existing partitions of C^2 systems generally require 2K to 10K words of memory per core.

5. External communication per process is between 50 and 20,000 words per second.

6. Internal communication between processes within a system varies between 1K and 5K words per second.

The design of C^2 systems must proceed from the point of view of the total combat scenario; the systems integration aspect should be consumed within the overall design process.

SURFACE COMBATANT SHIP COMBAT SYSTEMS ARCHITECTURE
R. P. Cullen (NSWC)

A letter issued by Vice Admiral Doyle, the Deputy Chief of Naval Operations and Surface Warfare, dated 3 November 1976 provided broad operational guidelines for future combat systems designs. Summarizing very briefly, the letter stresses:

1. Delegation of authority to warfare areas

2. Simultaneous or independent action in all warfare areas

3. Retention of overall control by command

The structures of existing combat systems on surface ships are largely accidental, and they do not agree with the philosophy presented in the letter. In a typical command organization, system-level functions are generic rather than being related to warfare areas, which makes sharing of sensors and sensor information difficult. The systems are not very flexible, and interconnection problems become more complex as the systems expand.

The type of guidance presented in Vice Admiral Doyle's letter was badly needed. We are presently developing design requirements based on the letter. The Navy is working towards organizing the distribution of its combat systems to make them more efficient and more flexible.

THE APPLICATION OF STRUCTURES DESIGN AND DISTRIBUTED TECHNIQUES TO AVIONICS INFORMATION PROCESSING ARCHITECTURES

Louis A. Naglak (NADC)

A structured-designs procedure based upon functional-mode partitioning can optimize the advantages offered by distributed-processing techniques. Core avionics, which comprise processing systems networks and navigation and communication systems, permits transportability among aircraft for various missions.

The structured-designs technique breaks down subsystems into functions and assigns these functions to hardware, software, and firmware components. A system functional design is applied to the variables involved in a distributed computer network to determine which functions to be implemented in software, hardware, or firmware. These variables are:

1. The selection of resources

2. System module interfaces

3. Interaction between modules

The distribution of processors within subsystems is guided by structural and aircraft systems designs constraints. Software can be simplified by using standard languages, software support, and standard interfaces in software and hardware.

Avionics information processing deals with the following subsystems:

1. Communications

2. Radar

3. Navigation

4. Acoustics

5. Optical

6. Magnetics

7. Display subsystems

8. Operator interfaces

9. Storage

SEAMOD

Don Eddington (NOSC)

The SEAMOD studies examined the modularity concept for shipboard sensors and weapons systems and found that distributed processing in this application can achieve:

1. Greater survivability

2. Savings in life cycle costs of software and hardware

3. More frequent modernizations in less time

Distributed processing was selected as the solution to two major problems of shipboard central data systems:

1. The major decoupling required by subsystem changeouts

2. The integration of new software into the central system

The goals set for SEAMOD were (1) to examine the data organization of combat direction systems and (2) to partition functions by requirements and not by hardware. The primary study showed that a combination of data buses could accommodate a 30-year life cycle of "changeouts" on a centralized combat direction system, with substantial software savings. A second study developed a model of a distributed system based on the data requirements of each process. Centralized systems programs were partitioned into 800 to 1000 functions, and these functions were repartitioned according to mission, configuration, subsystem, and implementation dependencies. These were assigned to a larger number of smaller, cheaper, and sometimes slower computers. The various processors (up to 50) were tied together by common data bases. Even though the total system costs about the same to build as a centralized system, over the life cycle of the ship approximately $8 million was saved because changeouts were cheaper. The model was later extended into a generic combat directions system, and it is presently being applied to the FFG-X.

INTERCONNECTION TECHNOLOGY AND NAVY SYSTEMS

Allan Clearwaters (NUSC)

The Navy is a systems-oriented organization. The use of distributed processing in Navy systems must be evaluated in terms of its impact on the total system operation. In that frame of reference, the greatest

asset offered by distributive processing is its flexibility. Problems created by expanding technologies and long-lead times can be counteracted by systems that can absorb technological change.

Two key concepts in designing flexible systems are:

1. Functional design

2. Flexible interconnection architecture

A function has two definitions within a system, each one subject to changes at different rates. The logical definition describes what the function does; the physical definition describes the actual incarnation of the function and will change more rapidly than the logical definitions. Technological updates to systems designed to accommodate changes in their physical functions would not be a major problem.

Interconnection technology must provide a system architecture capable of absorbing functional changes. This means, in effect, that implementation must also be functional. The tight functional interconnections in current systems preclude the incorporation of state-of-the-art technology into those systems.

OVERVIEW OF RESEARCH PROGRAMS IN DISTRIBUTED COMPUTING

Leonard S. Haynes (ONR)

The Office of Naval Research (ONR), Code 437, supports many basic research projects relevant to both loosely and tightly coupled distributed computer systems. The Office isolates key problems causing bottlenecks in the development of new systems for the Navy and provides funding in support of research leading to their solution. Basic issues currently being supported concern resource management and operating systems design, reliability, and security. A brief description of specific projects will provide an overview of current efforts:

1. Dr. Kimbleton, University of Southern California. Queing models and dynamic resource allocation

2. Professor Lienz, UCLA. Performance assessment using an aircraft carrier command-control system

3. Professor Van Dam, Brown University. Dynamic distributed processing, effectiveness under various forms of data structure segmentation

4. Professor Chu, UCLA. Dynamically reconfigurable and survivable networks

5. Professor Morgan, University of Pennsylvania. Distributive-data bases and automatic alerting on the occurrence of specific information patterns

6. Professor Hsiao, Ohio State. Design of a secure, efficient data-base computer

7. Dr. Carl Hewitt, MIT. Electronic messages carrying algorithms to be executed on their receipt

8. Professor Farber, UC Irvine. A cable-based ring network for inter-connecting processors

9. Professor Lipton, Yale. Synchronization primitives and formal math models of parallel processes

ONR is also funding, for the second year, a workshop on Distributed Processing at Brown University.

A COMMAND CONTROL CONCEPT
John Griffin (EG&G)

Over the years the trend has been that the systems acquisition manager has been able to buy more and more performance for the dollar; there has been an order-of-magnitude improvement every five to ten years. However, he has not really been able to see the benefits of lower hard-ware costs. Actually, the sharply escalating software costs are depriving him of total performance savings.

The price of a box of electronics has remained fairly constant over the past 20 years, but systems capabilities and systems performance have been increasing rapidly. Things can be done faster than before, but more things and more complicated things are being done.

From an analysis of the differences in programs and types of pro-gramming (is it a control problem or a data-base problem, is it slightly different or a whole lot different from a previous job, are the programmers new or are they experienced, etc.), a complexity factor can be arrived at. The average productivity for military systems is 100 instructions a man month, and the cost of programmers to the Navy is about $66,000 per year--about $55 per instruction. Thus, a 100,000-instruction pro-gram--which is not a very big program--costs $5-1/2 million.

Development costs are only part of the story. Support and maintenance cost much more. If the system has a 10-year life cycle, development represents only about 30% of total cost.

In the future, systems will continue to get bigger, more complex, and therefore more expensive. In the past, centralized architectures or closely coupled architectures were developed as a measure to overcome the high cost of hardware. This method of doing things, however, has required much more coordination in terms of systems integration, adding a lot of dollars and a lot of time. Some development cycles are as long as 15 years.

A solution to this problem must be based on the development of well-defined interfaces, following good systems engineering practices. Some suggested basic principles are:

1. Well-defined interfaces between functional groupings of hardware

2. Well-defined systems functional requirements

3. Independent design and analysis responsibilities for each system

4. Standard data transmission formats between systems

5. System-level design and implementation ground rules

6. A simple method of altering interfaces and requirements

DISTRIBUTED COMPUTER INVESTIGATIONS FOR SHIPBOARD COMMAND CONTROL SYSTEMS APPLICATIONS

E. J. Wells (NOSC)

NOSC is presently investigating two different share-bus distributive processing concepts for a combat-direction system. The specific application is for a real-time, computerized-tactical information system providing automated command and decision-making functions for the control and coordination of sensors and weapons systems. The two architectures presently being investigated are (1) the shipboard data multiplex system (SDMS) and (2) the Honeywell experimental distributed processor (XDP). The SDMS is comprised of five buses, each with a centralized controller. The XDP, however, has a single bus with decentralized control.

The five physical buses in SDMS run the length of the ship. Each has its own continuously scanning traffic controller. There are four

data channels and one control channel for each bus, making 20 data channels in all. The control channel polls areas and then informs that messages may be transmitted.

Area multiplexers connect remote multiplexers, and thus user equipment, to the buses and are physically connected to all five buses. This enhances system revivability. Also, all user elements connected to SDMS have two entirely separate physical paths to the bus.

There are 13 different types of I/O cards supportable by SDMS, including interfaces for displays, guns, missile launchers, and, our main concern, processors. The entire interface capacity of the system is 32 remote multiplexers x 64 I/O cards per remote multiplexer.

SDMS presently has a two-processor configuration but will expand in the near future to five processors. Three different processors will be used in the configuration: the 1215 and L20 military computers and the ALPHA LSI minicomputer. The ALPHA minicomputer will also be employed by the XDP, enabling the use of some standard applications software for both systems.

The XDP system was discussed in a previous presentation. Briefly, this system is characterized by its decentralized bus control. All processing elements are connected to a global bus by a bus interface unit (BIU), which has no location constraints as to where it can tie onto the bus. A 256-bit-time-lock vector in the BIU allows it to determine when its processing element has control of the bus.

AVIONIC INFORMATION PROCESSING SYSTEMS DESIGNS METHODOLOGY

Stanley B. Greenberg (NADC)

NAVAIRDEVCEN, in conjunction with Sperry Univac, has been developing a unified design methodology for avionic information processing systems. The methodology, presented briefly in the following list, provides a structured, step-by-step guideline for the design process.

1. Define the application in terms of functional-performance requirements

2. Break down the requirements into functional units called decompositional elements

3. Identify interfaces between these elements, using them to map elements into larger units called decompositional units (DUs)

4. Build a control structure with the DUs

5. Select an implementation medium (hardware, software, firmware) for each DU

6. Select hardware for each component (standard or nonstandard)

7. Configure hardware system

8. Map other DUs onto the hardware system

9. Validate the design with a system simulation

At each step in the process, a decision matrix is used that contains appropriate assessment criteria. The decision matrices presently existing are qualitative, with decision elements being positive, neutral, or negative. They are also preliminary, and work is continuing to quantify the matrix elements and to further screen the assessment criteria. Expected future improvements to the methodology include the incorporation of existing design tools such as the University of Michigan's User Requirements Language Analyzer, strategy assignment, and the automation of selected procedures.

MONITORING A DISTRIBUTED PROCESSING SYSTEM

Richard Fryer (NWC)

At the Naval Weapons Center in China Lake, California, a software validation and control system (SOVAC) has been evolving over the past 10 years as a response to our needs in performing tactical avionic checkouts. The first system, a memory-bus monitor, was too manual to be of much use. The incorporation of minicomputers in the memory-bus monitor produced a second-generation test device that is highly utilized by the avionics industry and the Air Force, known as the computer monitor and controller (CMAC). The SOVAC is the third generation system.

The SOVAC examines the system's buses (I/O bus and automatic ground equipment bus) via a custom interface. A microcontroller is tied to the interface, which performs the real-time access requested by SOVAC users. A microcomputer (LSI 11) and a graphic terminal (Tektronic 4014) complete the system. This is a testing device for a single computer. The next stage in its development will be an adaptation so that it can be used on multiple computer systems.

The SOVAC user language is easy to use, adapted from BASIC and other successful languages. The command set has a logical sense to

it that is easily understood. Editing can be performed in the user mode. Commands without line numbers are executed in direct mode.

The complex-condition command (IF COMPLEX CONDITION IS MET THEN ACTION) detects three forms of complex conditions. The user can combine these to make more complex conditions. The complex condition device gives the user the ability to follow his intuitions about the cause of a problem to understand exactly what is happening within the system so that he can trace the bug.

A SOVAC for multiple computing systems will extend the complex-condition capability to allow the introduction of external events (events from another CPU, for example). With this fourth complex condition form, the user will identify an outside monitor and an event within that monitor (optional) and SOVAC will:

Determine the effect of the chosen event on the processor being tested, through looping, logical ands and ors, and time synchronization.

We are presently working on an application of this technique to be used on the F-18 multiplex system.

A MULTI-DISTRIBUTED PROCESSOR SYSTEM (MDPS)

William Sheppard (NOSC)

NOSC has developed a multi-distributed processor system (MDPS) capable of supporting 24 processor modules per cabinet. The MDPS cabinets can be connected together via an interconnecting network that essentially allows any processor module in any cabinet to communicate directly with any other processor module in another cabinet. The I/O processor modules are designed to interface between the exchange bus on which all processor modules communicate and peripheral devices or communication links external to the MDPS. The computing processor modules have no I/O other than via the exchange bus, and their main function is to process data that has been entered into the MDPS via an I/O processor module.

The MDPS was to be installed in September 1977 as an operational system at the Strategic Air Command (SAC) Headquarters, Offutt Air Force Base, Omaha, Nebraska. The MDPS was to be implemented as a message-routing switch for the program-assisted console evaluation resource (PACER) system. The MDPS was to connect eight 50-K-bit communication channels from two Honeywell 6080 mainframe computers to 48 different display terminals in support of the real-time PACER operation.

Another proposed utilization of the MDPS architecture is as an interface unit to the AUTODIN switching network. This AUTODIN interface would allow various processors to communicate with AUTODIN via their standard communication channels with no software or hardware changes to their systems.

Experience with the MDPS architecture indicates that this architecture is a viable solution to many Navy problems, such as the instrumentation of the Electromagnetic Vulnerability Assessment Project (EMVAP) and the automated message entry systems (AMES).

FLOW GRAPHS AND DATA FLOW GRAPHS USED TO AID THE PARTITIONING PROCESS

Uno R. Kodres (NPGS)

A diagram of a standard electrical network can be expressed in flow-chart form, and therefore techniques for solving voltage and current problems in a standard electrical network can be adapted to solve execution-time unknowns for software programs. A directed graph, with arcs representing resistors, current generators, and voltage sources, is used to solve for current or voltage unknowns in an electrical network. Relationships between these arcs are then developed, Kirkoff's laws are applied, and equations are developed expressing total current and voltage in terms of known quantities.

Applying this technique to execution time analyses in software programs, a directed graph is drawn from a flow chart. The arcs represent execution instruction sequences. Kirkoff's laws are adapted to this situation by substituting time values for each execution sequence for the resistance values used in the electrical problem. The total execution time for a program is the sum of all the matrices representing each arc in the directed graph.

A problem in distributed processing concerns partitioning. Data flow graphs, which illustrate the inputs and outputs of a system in terms of data and functional quantities, are useful tools in the partitioning process. One graph describes a single operational interval, with definite starting and terminal points. In this manner, a complex flow chart can be represented by a series of individual graphs of data-flow sequences. The partitioning problem is thus reduced to a graph-partitioning problem, for which there are many effective algorithms minimizing the number of interconnections between parts of the graph. This will effectively reduce the number of communications requirements between the system's computers.

Data flow graphs are also useful in analyzing flows for the purpose of testing programs. They indicate which execution sequences are independent from each other, which reduces the number of combinations of sequences to be tested.

SECNAV POLICY GUIDANCE

B. Zempolich (NAVAIR)

A well-conceived policy can maximize competitive procurements, minimize loss of previous investments, and allow new-technology investments. This policy should also establish procurement flexibility by weapons systems and by platform. Standardization of interface characteristics is a must.

PROTOCOLS

Paul Levine (NUSC)

The primary reason for intersystem connection is to pass information. A strategy common to computer networking efforts is the separation of information handling into several distinct functional levels. These levels correspond to the conceptual views the networking machinery has of information at different stages of its operation. This formal layering is motivated by a desire to support functional abstraction. Modular programming allows each member of a software team to develop an operational code independently of the language and style of the code with which it must interact. Similarly, layers of abstraction for information exchange allow a network implementer to build each of the several functions of a network out of any of the appropriate and available technologies. The idea is to be able to modify any layer of a network without disturbing any other layers. Such modifications may become desirable as a result of changing technologies or requirements.

Six levels of function and therefore six layers of protocol that support information exchange have been identified. These range from the electrical level at which information is viewed as pulses of voltage or light to the highest level at which information is a medium for interprocess communication. At each level, we have identified particular information processing functions to be performed. While not every facility at each level is necessary for every network implementation, the clear separation of functions into layers allows simpler modification and maintenance of the network environment.

A preliminary classification has been made of some of the more widely discussed protocols; however, a much more detailed examination of existing protocols is in order. Further, the time is nearing when the selection of military (if not general) network protocols will become crucial to systems designs and construction. With the proper review and evaluation of existing (and proposed) computer networking protocols, a set of standards can be developed to meet the increasing desire to interconnect computers and computer-based systems in a reasonable time frame.

ADAPTABLE SHIPBOARD TACTICAL DATA DISTRIBUTION SYSTEM

LCDR H. C. Schleicher (NOSC)

The interconnection of computer components using bulky, expensive, parallel cabling and manual switches has been replaced in shore-based complexes at three locations by parallel serial converters, electronic matrix switches, and serial parallel converters. The added versatility of the matrix switch is as desirable as its savings in material, weight, and installation costs. Also, automatic fault detection and correction are feasible with the matrix switch.

Specific problems existing in current shipboard systems were identified as follows:

1. No pooling or sharing of backup equipment

2. Slow recovery time/reconfiguration time

3. No real configuration management

4. Physical size and weight of parallel cables and switches

5. No facility for growth and expansion

The adaptable shipboard tactical data distribution system was designed to provide for these deficiencies. The system is compartmentized and functionally expandable, reliable through redundancy, and self-healing. Future tactical data systems, both shipboard and shore-based, will be impacted by the necessity to conduct serial interchange of data through a versatile, compartmentized, expandable distributed switching and control system.

IEEE STANDARD 488
Jim Howard (UCSB)

IEEE STD-488 defines a general-purpose digital interface for testing instruments of varying complexity. It permits the configuration of a multilevel system without a major design effort, providing a standardized communication link between the components of the system. A brief summary of the specification is presented here.

The maximum number of devices that can be interconnected on one bus is 15. The interconnection network is either star or linear. There are 16 active signal lines. Message transfer is byte serial, bit parallel, and asynchronous. Data is transferred with an interlocked, three-wire handshake technique, with a maximum rate of 1 megabyte per second over a limited distance of 2 meters. There is a total of 10 interface functions. The primary address scheme accommodates 31 talk and 31 listen; the secondary scheme expands the address to 961 talk and 961 listen. Controllers can pass control to each other, but only one controller can be active at a time. The driver and receiver circuits are TTL compatible.

IEEE STD-488 specifies eight data lines, three data-transfer lines, and five active-bus-management lines. The cable connection strategy is well defined, and the cables themselves are standardized.

Specific advantages of IEEE STD-488 are:

1. It has inexpensive configuration with off-the-shelf instruments.
2. Flexibility allows easy reconfigurations.
3. Many vendors support it.
4. It is nationally accepted and becoming international.
5. Instrumentation interfaces are easy to design.
6. Potential exists for distributed systems.
7. It is mechanically and electrically standardized.
8. Data lines can be increased without changing protocols.
9. Success promises a 488 LSI interface chip.

Some limitations of the standard, besides those built in by the specifications described above, are:

1. Bus transfer rate is limited by slowest listener in system.

2. Implementation is difficult, relatively, because of the specifications's complexity.

3. Address assignments for various classes of instruments have not been standardized.

4. Software has not been standardized.

NATIONAL I/O INTERFACE STANDARDS

Dell Shoemaker (GSA)

In 1967, the American National Standards Institute (ANSI) organized a study group devoted to I/O interfaces. The effort to establish an acceptable standard continued and is now in the final stages of the review process.

The proposed standard concerns lower-level interfaces, with "lower level" defined as device-independent interfaces, basically simpler and cheaper than channels, directed towards a specific application. The standard does not include interfaces appropriate for distributed systems, but those are in the development stage. Also, two standards for minicomputers will be distributed for public comment within three to six months.

The entire project has been controversial, and the standard has withstood much criticism. Most negative comments have been directed towards the fact that a specific manufacturer's product is being supported by the standard, in contrast to IEEE STD-488, which does not lend itself to any particular manufacturer.

The proposal is out for the comment period at present. When that is over, the technical committee will revise or modify the document, if necessary to achieve acceptance, as dictated by industry consensus. The X3 Committee, the final technical committee at ANSI, must approve the proposal, and then it will be presented to the Board of Standard Reviews for final approval.

MIL-STD 1553
Richard De Sipio (NADC)

The 1553 protocol was generated by the Air Force a few decades ago to prevent the proliferation of data-link multiplex systems. The goal was to facilitate the interfacing in central integrated systems. At the same time, the Navy developed a general-purpose multiplex system based on the SDMS holding architecture, which was appropriate for distributive systems. The major difference between the 1553 and the Navy's system was that the Navy's system included the capability of handling off-bus control, and the 1553 did not. A coordinated effort between the triservices, DOD, and industry resulted in the 1553A, which included a dynamic bus-allocation feature, and also a stand-alone controller feature and a free-running polling bus controller. A 1553B, which provides for a broadcast mode, is now being generated.

Standardization is a slow process, but with patience and proper planning standards can guide industry-developed technology in desirable directions. A particularly useful planning technique is partitioning, which permitted the standard to accommodate new technologies as they developed. The 1553 was applied first to aircraft-control-status signals. Later, in the F16 and F18, it was applied to all signal types. A program is presently under development for an audio-multiplex system, which will use the 1553A. Other wide-band requirements are also being investigated, all to use the 1553A protocol. NADC convinced industry that we would be continuing to use this protocol and expanding its applications, and they developed LSI chips to satisfy the protocol.

This development was never contracted for; it evolved because industry was assured, through NADC's pattern of development, that the military would purchase a technology that accommodated the 1553.

BUSES AND DISTRIBUTIVE COMPUTER SYSTEMS
David Rennels (JPL)

A complexity problem became apparent during the design of a distributed processing system for future spacecraft applications. The system consisted essentially of six spacecraft-subsystem terminal modules interacting with two higher-level processors, with interactions too complex for one removed from the design process to understand. It has been realized that a distributed-processing system too complicated to test and understand will not sell.

The configuration was greatly simplified by eliminating all mad interrupts and constraining the I/O to discrete points in time, with software for each module running in incremental segments. The result was 250 words of sequence programming in each module and a global-command process tying the system together.

Desirable features for the system's bus are:

1. Transparency. DMA-block transfers should be by name and not by address.

2. Simplified formats. The communications within the computer should be limited to only that necessary for the bus to transfer data. Status messages are desirable for reliability.

3. Limited complexity. System users should be able to understand the interfaces.

4. Type timing. Requirements are not necessary.

5. Reliability and fault tolerance. The higher-level modules must be fault tolerant. This system will use a daisy-chain interconnection scheme to achieve redundancy.

JPL designed their own bus and used the 1553-type interface, with a few augmentations. Currently, the use of microprogrammed-data terminals instead of data path is being studied, which would allow for flexible DMA channels interfacing with a number of microprocessors. A possible extension of this configuration is the inclusion of more buses and bus interface hardware. A further step is the incorporation of a uniform set of data paths and a bus terminal into a chip.

In conjunction with NASA, JPL is also examining the issue of fault tolerant distributive computer systems, which is essentially a stand-by redundant system at the higher level. Although fault tolerance at the intelligent-terminal level may not always be necessary, there are cases (i.e., attitude control) when redundancy is desirable.

A FAMILY OF SPECIAL-PURPOSE PROCESSORS FOR DISTRIBUTED-DEDICATED COMPUTER SYSTEMS

Maniel Vineberg (NOSC)

A family of dedicated, special-purpose processors called the programmable algorithm machine (PAM) is being developed. The PAM will feature a processor composed of multiple-processing elements, separate instruction and operand memories, and instruction pipelining. It is designed to execute efficiently over a class of algorithms that exhibit (1) a high frequency of independent operations and (2) a low frequency of branching.

The PAM will normally be programmed in an algorithmic language (e.g., a subset of ALGOL) but will also be programmable in the PAM assembly language (PAL). Each PAL instruction includes a postfix assignment part and a sequencing part (optional).

The PAM will be supported by a unified software system consisting of a compiler, a parameterized assembler, and a parameterized simulation. Compilation, assembly, and simulation time are less important in the dedicated environment of the PAM than is execution efficiency. Therefore, the PAM software will be used to optimize PAM application programs and to verify and measure the performance of those programs on various versions of the PAM in order to produce a PAM version tailored to the application.

Once optimization is complete, the actual PAM hardware version will be assembled. As a dedicated special-purpose processor, the PAM will perform preprogrammed, preoptimized algorithms at the request of a controlling device. Versions of the PAM, tailored to specific applications, will perform time-critical functions now performed by costly fixed-program hardware.

Jim Campbell (Office of ASN, R&D)

The Navy's requirements for I/O standard control interfaces and standard internal-bus structures and memory protocols are based on the promises of near-term technology. The Navy will soon have chip or programmable chip protocols, but will not be able to implement them if there is no horizontal integration between its systems. Standardized intersystem communications must be accepted if advantage is to be taken of future technologies.

Bob Gordon (NUSC)

The military is so preoccupied with the concept of real time that its system trade-offs are approximately 80% performance--only about 15% reliability and 5% flexibility. Commercial systems, on the other hand, are trading off 50% performance, 20% reliability, and 30% flexibility. This is the key to the military's problems concerning obsolete technology and high system-support costs. The Navy needs to develop a strategy that would allow technology improvements to maximize competitive procurements and minimize the loss of previous investments.

Phil Andrews (NAVSEA)

Distributed processing, or any other technological innovation, is only as good as the ability to use it. The Navy's present management philosophy is a limiting factor on the use of distributed processing. After being assured that distributive processing can help us come closer to overall goals, then greater support must be given to it. This new technology must be implemented in systems now under development.

In developing distributed processing systems, a top-down approach must be used and the total platform considered as a system. Military systems must be fault tolerant and reconfigurable, and, since improvements are inevitable, they must have the ability to adapt to new technologies.

Future shipboard systems will probably be hybrids, as opposed to true distributed processing. However, before the Navy's systems will be capable of truly performing their missions, there will have to be some changes in the entire management structure as it is known today.

Charles Arnold (NUSC)

Operating system issues for distributive systems include:

1. Partitioning, which should be the concern of applications engineers

2. Fault tolerance, which calls for dynamic reallocation capabilities and, therefore, centralized control. No one knows how to decentralize control.

3. Protection, which seems to be a subject that no one wants to discuss

4. Synchronization. To prevent deadlocks, there must be communication between functional subsystems.

As a representative of the software community, software-to-software and exec-to-exec interfaces need to be standardized.

Tom Wolff (UNIVAC)

One of the primary design objectives for tactical systems is adaptability to modifications. The capability to extend and modify a system without disproportionate costs and side effects can be achieved through modularity. Modularity cannot be controlled without structure. Structure implies standard interfaces, which implies a standard control philosophy--a system standard. Standards, however, must be good; otherwise, they will not be adopted and they will not achieve their objectives.

As systems grow in complexity, the demands for the purely overhead functions of communication and control increase. Problems occur when systems are designed to spend a disproportionate amount of time in accomplishing overhead. There is no such thing as a best interconnect; "the best" will depend upon specific environments, objectives, and applications. Our standards must permit taking advantage of all viable interconnects.

The Navy has had standard hardware-to-hardware interfaces for quite a while, with MIL STD-1397. It is an effective interconnect, and the Navy cannot hope to achieve anything better concerning hardware-to-hardware interfaces. The Navy has ignored, however, interfaces between software modules and exec-to-exec types. At this point in time, the

integration of improved interface capabilities in tactical systems depends upon the standardization of these other interface types.

A big difference exists between the objectives of tactical systems and the typical commercial system. The standards must reflect those differences.

Carl Mattes (NADC)

This three-day workshop can be summarized in four words: challenging, in that distributed processing holds promises for many improved capabilities; confusing, in that participants cannot even agree on a definition; frustrating, in that one does not yet know how to achieve some of the capabilities described as necessary for true distributed processing; and encouraging because several Navy labs are currently developing distributed processing systems. Attendees have all achieved a basic understanding of the state of the art, examined the Navy's objectives and approaches for reaching its goals, and understanding many of the associated problem issues, although there was no talk about how to resolve them. There was discussion on the need for standardization and flexibility in Navy systems.

It is recommended that, for the future, program managers develop a set of definitions for distributive processing and another set for their platform requirements. The most important of those requirements should be chosen for which there is no capability today, and the R&D budget should be used to develop them. Our near-term systems must be continuously improved with the existing knowledge and not hold back for the ultimate system. At the same time, the ideas that promise the greatest return on investments for the long-term future can continue to be developed.

John Machado (NAVELEX)

The original letter describing the workshop listed five objectives that could be used as a basis for summarizing what was accomplished.

1. Understand the objectives of distributed systems. There is a definitional problem, and there was agreement that a set of common goals for distributed systems be developed. From the systems design aspect, a distributed system is one implementation technique that can be used to satisfy goals such as reliability and flexibility.

2. Review current Navy approaches. The Navy's distributed systems are mostly in the R&D stage, in the 6-1 and 6-2 areas. However, the Navy does have dispersed systems in the form of dedicated microprocessors and microcomputers. The Navy is experiencing difficulties in transferring technology to the field because of the lack of existence proofs.

3. Survey the state of the art. Navy laboratories and industry are far apart in this area.

4. Common issues. Protocols, network architectures, and system trade-offs were discussed in much detail.

5. Trends, ideas, technologies, and their impact or potential payoffs. The virtualization of networks seems to be a goal that most are working for. New technologies kept transparent to the programmer is a necessary goal; otherwise, systems will continue to increase in complexity.

An understanding of the state of the art is essential to any discussion. Even though the issue of standardization is not new, it cannot really be talked about until there is an understanding of the state of the art. Also, several different standardized protocols are needed, each to be used for different implementations.

Walt Warner (NSWC)

Applications programming is made easier by making systems software more complex. Since system software is a one-time development, that is the proper place to hide the systems complexities; however, engineers must not forget that every time they build flexibility or generality into a system, they are complicating the job of the system programmers. Since the amount of software overhead is already a real problem, the situation calls for more cooperation between engineers and computer scientists.

Another problem deserving cooperation is fault detection. Planning must be made for the situation in which a programmer has to solve a failure in a system he did not write. This calls for failure isolation and software verification/validation capabilities.

A basic concern about standardization is that the Navy cannot expect to produce standards that everyone will interpret in the same manner if no agreement has been made upon standard definitions. Individual biases and goals must be forgotten and all must work together to develop a system that will operate for the men in the fleet.

GENERAL COMMENTS TO SESSION IV

Joseph Carter (NOSC)

There are certain functions in our systems that have priority importance, ones that cause real problems if they are down. There are others that are not quite so important. A system whose functions are tied together so that they go down in a realistic sequence; i.e., a graceful degradation as opposed to a catastrophic one would be ideal. Additionally, is it necessary for a UYK-7 to perform functions that could be handled by a microprocessor? There are other overkill conditions where hardware being used is too sophisticated for the job.

The Navy's systems should be specialized only to the extent necessary to do the job. To try to optimize everything is wrong, and standards should reflect the consideration that too many optimizations in one system cannot be implemented.

Richard West (NSWC)

This workshop has pushed the issues of improving performance and reducing costs of distributed systems into the background in favor of standardization. Those priorities are wrong. As evidence, consider that (1) the percentage of our electronic systems budget spent on support costs is increasing every year, and (2) our existing systems do not perform.

Most complaints about existing systems concern their lack of testability and maintainability. However, some of our standardization goals seem to contradict our professed testability and maintainability objectives.

APPENDIX A

ATTENDEES

ATTENDEES

1. Aarons, Lee, OP 0980, OPNAV, Washington, DC 20550
2. Andrews, Phil, NAVSEA 03416, Washington, DC 20367
3. Arnold, Charles, Code 313, NUSC, New London, CT 06320
4. Ball, Dan, Telecom System Incorporated, 2300 South Ninth Street, Arlington, VA 22204
5. Bernstein, George, Code 04310, NAVSUPSYSCOM, Washington, DC 20376
6. Berry, Roger B., Code 6105B, NSEC, Washington, DC 20362
7. Bettis, Al, UNIVAC, 1901 North Moore Street, Arlington, VA 22209
8. Billet, Wendy, Code PME108, NAVELEX, Washington, DC 20360
9. Bock, E., Code 551, Naval Air Development Center, Warminster, PA 19874
10. Buttinger, James, LCDR, Code 54BK, NGPS, Monterey, CA 93940
11. Campbell, Jim, Office of ASN(R&D), Washington, DC 20350
12. Carlock, Jack C., Code K502, NSWC, Dahlgren, VA 22448
13. Carter, J. W., Code 8332, NOSC, San Diego, CA 92152
14. Clarke, Ed, LT, Code F36, NSWC, Dahlgren, VA 22448
15. Clearwaters, Allan, Code 52143, NUSC, New London, CT 06320
16. Coffey, Ted, Data Solutions Corporation, 2095 Cambridge Road, Vienna, VA 22180
17. Colby, Dave, Code DD, NSWC, Dahlgren, VA 22448
18. Connelly, D. J., Technical Director, NAVDAC, Bldg. 166, Washington, DC 20374
19. Converse, Robert A., Code 3521, NUSC, Newport, RI 20840
20. Cooper, D. L., CAPT, NAVSEA (Code 06C)/NAVSEC (Code 6170), Washington, DC 20360
21. Cooper, Jack, CDR, MAT 09Y, Naval Material Command, Washington, DC 20360
22. Cottrell, James, Code WU-32, NSWC, White Oak, Silver Springs, MD 20910
23. Crisp, Harry, F36, NSWC/DL, Dahlgren, VA 22448
24. Cudd, George, Code AR 52022, Naval Air System Command, Washington, DC 20361
25. Cullen, R. P., Code N30A, NSWC, Dahlgren, VA 22448
26. Dean, Edward, Code WA-54, NSWC, White Oak, Silver Springs, MD 20910
27. Dejka, Bill, Code 7503, NOSC, San Diego, CA 92152
28. De Sipio, Richard, EC-5491, NADC, Warminster, PA 19874
29. Dietzler, LCDR, Code 08T212, HQ NAVMAT, Washington, DC 20360
30. Doracek, Dr. Joseph, IBM, FSD, Owego, Tioga Co., NY 13820
31. Duncan, Daniel, D. A., Code K503, NSWC, Dahlgren, VA 22448
32. Eddington, Don, Code 732, NOSC, San Diego, CA 92152
33. Enslow, Dr. Phillip H., Jr., Georgia Institute of Technology, Atlanta, GA 30332
34. Eyres, Russ, Code 822, NOSC, San Diego, CA 92152
35. Faucher, K. A., Code 8245, NOSC, San Diego, CA 92152

ATTENDEES (Continued)

36. Franklin, Jeff, Code WA-54, NSWC, White Oak, Silver Springs, MD 20910
37. Fratilla, R., ELEX 330, NAVELEX, Washington, DC 20360
38. Fredette, R. C., Deputy Director, NAVDAC, Bldg 166, Washington, DC 20374
39. Fryer, Richard, Code 31, NWC, China Lake, CA 93555
40. Gage, Doug, Code 732, NOSC, San Diego, CA 92152
41. Gilcrease, David, Code WA-54, NSWC, White Oak, Silver Springs, MD 20910
42. Gordon, Dr. Bob, Code S314, NUSC, New London, CT 06320
43. Greenberg, Stanley B., Code 2071, NADC, Warminster, PA 19874
44. Griffin, John, EG&G, WASC, 2150 Fields Road, Rockville, MD 20850
45. Hagen, Chuck, Code 824, NAFI, 6000 East 21st Street, Indianapolis, IN 46218
46. Hawkins, Robert D., NWC, China Lake, CA 93555
47. Haynes, Leonard S., Code 427, ONR, Arlington, VA 22217
48. Healy, Leonard, Code N214, Naval Training Equipment Center, Orlando, FL 32813
49. Hein, Robert R., Code N20, NSWC, Dahlgren, VA 22448
50. Hill, W. Henry, Code 00G, Naval Sea Systems Command, Washington, DC 20360
51. Howard, Jim, Dept. of EE and CS, University of California, Santa Barbara, CA 93106
52. Hunsucker, Johnny B., Code 732, NCSL, Panama City, FL 32401
53. Jarrett, B. C., Code F36, NSWC, Dahlgren, VA 22448
54. Jensen, E. Douglas, (MN17-23401), Honeywell, Inc., Systems & Research Center, 2600 Ridgeway Parkway, Minneapolis, MN 55413
55. Joeckel, Carl, NADC, Warminster, PA 19874
56. Johnson, Robert, MAJ, AFSC/XRF, Andrews Air Force Base, Washington, DC 20331
57. Kahane, Bob, ELEX 330, NAVELEX, Washington, DC 20360
58. Kasik, Ronald P., Code 4451, NUSC, Newport, RI 20840
59. Kesselman, Murray, RADC/ISCA, Griffiss Air Force Base, Rome, NY 13441
60. Kimbrell, Thomas, Code DN-20, NSWC, Dahlgren, VA 22448
61. Kodres, Uno R., Department of Computer Sciences, NPGS, Monterey, CA 93940
62. Kothstein, Thomas, Code 2071, NADC, Warminster, PA 19874
63. Larson, Ron, Code 570, NAVELEX, 6565 Arlington Blvd., Falls Church, VA 22042 Department 315
64. Lawson, Charles, Code 032, NAVSEA, Washington, DC 20376
65. Laybourne, Paul, N30, NSWC, Dahlgren, VA 22448
66. Letsinger, Robert, Code 931, NAFI, 6000 East 21st Street, Indianapolis, IN 46218
67. Levine, Paul, Code S3143, NUSC, New London, CT 06320
68. Liwski, P., NSEC, Washington, DC 20362

ATTENDEES (Continued)

69. Lomas, Leighton, Pentagon, AF/RDX, Washington, DC 20330
70. Lucas, Bob, Code WA-51, NSWC, White Oak, Silver Springs, MD 20910
71. Lucas, William, Code N20, NSWC, Dahlgren, VA 22448
72. MacDonald, Bill, Computer Science Corporation, 6565 Arlington Blvd, Falls Church, VA 22042
73. Machado, John, ELEX 330, NAVELEX, Washington, DC 20360
74. Mattes, Carl, Code 2071, NADC, Warminster, PA 19874
75. McCloud, K. L., Code F36, NSWC, Dahlgren, VA 22448
76. McGlone, Kent, Code K53, NSWC, Dahlgren, VA 22448
77. Meher, Jean L., Construction Battalion Center (FACSO, Code 18A1), Port Hueneme, CA 90343
78. Mitchell, Richard, Code 50P3A, NADC, Warminster, PA 19874
79. Morris, Alfred H., Jr., Code K74, NSWC, Dahlgren, VA 22448
80. Mudd, Don, Code 733, NOSC, San Diego, CA 92152
81. Musselman, Marty, MAT 031, NAVMAT, Washington, DC 20360
82. Naglak, Louis A., Mgr. Inf. Processing and Display Div., NADC, Warminster, PA 19874
83. Norr, Bill, Code 2072, NAVAIRDEVCEN, Warminster, PA 19874
84. Nowicki, Chester, Code 2031, NADC, Warminster, PA 19874
85. Owen, Thomas F., Code K61, NSWC, Dahlgren, VA 22448
86. Palatucci, Jerry, Code 2032, NADC, Warminster, PA 19874
87. Perry, John G., Jr., Code K74, NSWC, Dahlgren, VA 22448
88. Pohles, Bill, Code 5522, NADC, Warminster, PA 19874
89. Pollock, Raymond M., Code K503, NSWC, Dahlgren, VA 22448
90. Pond, Robert M., Code K61, NSWC, Dahlgren, VA 22448
91. Poole, Eric L., Code 0113B, NAVFACENGCOMHQ, Alexandria, VA 22332
92. Prindle, Frank, Code 552, NADC, Warminster, PA 19874
93. Proctor, Dr. John, Data Solutions Corporation, 2095 Cambridge Road, Vienna, VA 22180
94. Pryor, C. Nicholas, Technical Director, NUSC, New London, CT 06320
95. Rennels, David, Jet Propulsion Lab 198/136, 4800 Oak Grove Drive, Pasadena, CA 91103
96. Rhodes, James W., Code 0112A, NAVFACENGCOMHQ, Alexandria, VA 22332
97. Rinfret, Mark R., Code 4232, NUSC, Newport, RI 20840
98. Schaar, Brian, LCDR, MAT 09Y, Naval Material Command, Washington, DC 20360
99. Schauer, J., Director, Computer Program Development Directorate, NAVDAC, Bldg 166, Washington, DC 20374
100. Schleicher, H. C., LCDR, Code 735, NOSC, San Diego, CA 92152
101. Schneller, Stephen A., Code 4451, NUSC, Newport, RI 20840
102. Sedgwick, Philip, Code 3524, NUSC, Newport, RI 20840
103. Sheppard, William, Code 732, NOSC, San Diego, CA 92152
104. Siewiorek, Dan, Carnegie-Mellen University, Apt. 506, Parkwood, 1300 Worchester, Framington, MA 01701

ATTENDEES (Continued)

105. Skullman, V. E. (AIR-53311B), Naval Air Systems Command, Washington, DC 20361
106. Smith, Bill, NRL, Washington, DC 20375
107. Sparbel, R., Code F26, NSWC, Dahlgren, VA 22448
108. Stemple, Eugene P., Code K61, NSWC, Dahlgren, VA 22448
109. Stern, George, Code CCIR, CMC, Washington, DC 20380, Marine Corps; Attn: Stern
110. Stoddard, Pat, CDC, 901 S. Highland Street, Arlington, VA 22204
111. Stout, Rex, ELEX 330, NAVELEX, Washington, DC 20360
112. Thombs, Herman W., Code K74, NSWC, Dahlgren, VA 22448
113. Tinklepaugh, Norm, Code 732, NOSC, San Diego, CA 92152
114. VanDamm, Andy, Box F, Computer Science, Brown University, Providence, RI 20912
115. Vineberg, Maniel, Code 732, NOSC, San Diego, CA 92152
116. Warner, Walt, Code K70, NSWC, Dahlgren, VA 22448
117. Wasem, Ronald G., Code DK-61, NSWC, Dahlgren, VA 22448
118. Weiss, Susan, Code 833, NAFI, 6000 East 21st Street, Indianapolis, IN 46218
119. Wells, E. J., Code 7333, NOSC, San Diego, CA 92152
120. West, Richard P., Code K61, NSWC, Dahlgren, VA 22448
121. Wilcox, James W., Computer Sciences Corporation, 5205 Leesburg Pike, Suite 800, Falls Church, VA 22041
122. Williamson, James, Code 552, NADC, Warminster, PA 19874
123. Willis, Larry, Code WU-32, NSWC, White Oak, Silver Springs, MD 20910
124. Wilson, Richard, CAPT, Code AIR533, NAVAIR, Washington, DC
125. Wolff, Tom, UNIVAC, UNIVAC Park, Box 3525, St. Paul, MN 55165 Mail Station U1U25
126. Wu, Y. S., Naval Research Lab, Washington, DC 20375
127. Yaroszewski, Ed, Code G10, NSWC, Dahlgren, VA 22448
128. Zempolich, Bernie, Air 360, NAVAIR, Washington, DC 20361